



Controlando errores por tipo

Transcripción

[00:00] Estamos ya en el punto ideal para hacer un último review de qué es lo que ha sucedido hasta ahora con este tema de las excepciones. Nuevamente en el mundo ideal, todo bien, tenemos la pila, tenemos el stack, dos métodos se van apilando, se ejecutan todo bien hasta que ocurre una excepción, una bomba. ¿Y qué pasa con esta bomba?

[00:23] Explota. Y si no sabemos cómo desactivar la bomba, si no le avisamos, a Java cómo puede desactivar la bomba, explota todo y la bomba hasta el final y el código nunca se termina de ejecutar. ¿Entonces, qué tenemos que hacer nosotros para poder lidiar con esta bomba?

[00:42] Avisarle a Java que puede explotar aquí una bomba y decirle cómo puede desactivar esa bomba. Entonces nuevamente aquí en el código, vamos a ver si de verdad está funcionando esto de la desactivación de la bomba. Aquí él está inicializando un string null y él está ejecutando un método de un objeto nulo.

[01:05] ¿Entonces él qué es lo que hace? Da un `NullPointerException` que es el nombre que tiene la excepción en Java. Recordemos que en Java las excepciones también son objetos, y como tal tienen atributos y tienen métodos. En este caso, tienen para hacerle un `getMessage`, puedo imprimir el `StackTrace`, puedo tener mucha información de la excepción debido a que es un objeto con propiedades y métodos también.

[01:34] Primera cosa. Entonces ahora yo tengo una duda, yo aquí estoy atrapando dos tipos de excepciones. ¿Qué pasaría? O no qué pasaría. ¿Pero será que de verdad él está atrapando la excepción correcta? Vamos a intentar algo aquí, voy a descomentar este código para saber que él está atrapando el error correcto, y aquí le voy a decir if, si i es igual a 3, cuando el índice sea 3, ejecuta esto de aquí.

[02:17] En pocas palabras, lanza este error de aquí. Si i es igual a 3, entonces tú me vas a lanzar un `ArithmeticException`, solo cuando sea 3. Si no es 3, me vas a crear esta `String` null y vas a lanzarme un `NullPointerException` y vamos a ver si de verdad está trabajando este catch con dos excepciones aquí.

[02:40] Entonces, nuevamente guardamos. Estamos en medio del debug del video anterior. Vamos a detener ese debug. Guardamos aquí. y le damos a ejecutar primero. Y vamos a ver aquí qué es lo que pasó. Mira. Vamos a ver todo lo que él imprimió. Inicio de main. Inicio del método 1, método 2, 1. Y atrapó una excepción.

[03:10] ¿Qué exception? `NullPointerException`. ¿En cuál índice? 1. En el 2, atrapó el exception. ¿Pero qué hizo? Lanzó un `NullPointerException`. Está bien. En el índice 3 atrapó exception y mensaje "by zero". ¿Por qué acá imprimió null? Porque en este caso este objeto, que es el `NullPointerException`, no tiene un mensaje, Java no configuró un mensaje para él. Entonces él imprime null.

[03:37] ¿Por qué? Porque es lo que sucedió. Null. Simplemente era un `NullPointerException`, no había nada ahí que imprimir, pero en el caso del `ArithmeticException` sí tiene una razón y es la división por cero, y el que imprime el Stack Trace de `ArithmeticException`. Y como él salió después de imprimir esa excepción, ya imprime la otra excepción también.

[04:02] Nuevamente en el 4 también atrapa la excepción y en el 5 finalmente pues nuevamente lanza su excepción. Fin del método 2 y acaba, fin de método 1, fin de main. Entonces, es de esta forma que ya podemos comenzar a trabajar

con los errores en Java. Nuevamente hacer hincapié en esto, los errores no son malos, es normal que sucedan errores en el código. El código perfecto no existe para nada.

[04:34] Lo que tenemos que saber es cómo lidiar con ellos, como nosotros podemos trabajar con eso y más importante aún, tenemos que aprender a leer el mensaje de la excepción. Si no sabemos leer el mensaje de la excepción, entonces ahí sí tenemos un pequeño limitante a la hora de explorar un error que está aconteciendo en producción o bueno, en el ambiente en que estemos trabajando.

[05:00] Entonces ya este es el primer paso, ya nuestro primer contacto fuerte ya con excepciones. En los siguientes vídeos, en las siguientes clases ya vamos a ver cómo crear nuestras propias excepciones, porque dependiendo el negocio también podemos crear nuestras propias excepciones.

[04:14] Por ejemplo, volviendo al ejemplo de de la clase cuenta, una excepción puede ser digamos no tiene saldo, excepción porque no tiene saldo, excepción por fraude, excepción por tarjeta bloqueada, etcétera, y nosotros creamos esas excepciones.

[05:30] ¿Para qué? Para decirle al sistema: "Si es que salió esta excepción haz esto de aquí, si no, haz esto de aquí, pero no te caigas, no dejes que el stack se rompa, termina tu ejecución". Ese es el propósito del control de errores. Nos vemos en la siguiente clase.